

10

```
FOR (i=0; i<2; i=i+1)
  begin
    out[i] = 8'b10101010;
    enable[i] = up[2*i];
  end
```

FIGURE 1

```
reg y [3:0];  
WHILE (x <= y)  
    begin  
        fpl_bit[x+y] = mm_iru[x-y];  
    end
```

The diagram shows two handwritten annotations. The number '13' has an arrow pointing to the variable 'y' in the condition 'x <= y' of the while loop. The number '12' has an arrow pointing to the expression 'x-y' in the assignment statement 'fpl_bit[x+y] = mm_iru[x-y];'.

FIGURE 2

16 18 20 14
↓ ↓ ↓ ↙
FOR (INIT; EXIT; INC)
begin 22
 BODY_OF_STATEMENTS;
end

The diagram shows a code snippet for a FOR loop. Handwritten numbers and arrows are used to label parts of the code: '16' points to 'INIT', '18' points to 'EXIT', '20' points to 'INC', '14' points to the closing parenthesis of the loop header, '22' points to the 'begin' keyword, and '20' points to the 'end' keyword.

FIGURE 3

```
out[0] = 8'b10101010;  
enable[0] = ~up[0];  
  
out[0] = 8'b10101010;  
enable[0] = ~up[2];
```




FIGURE 4

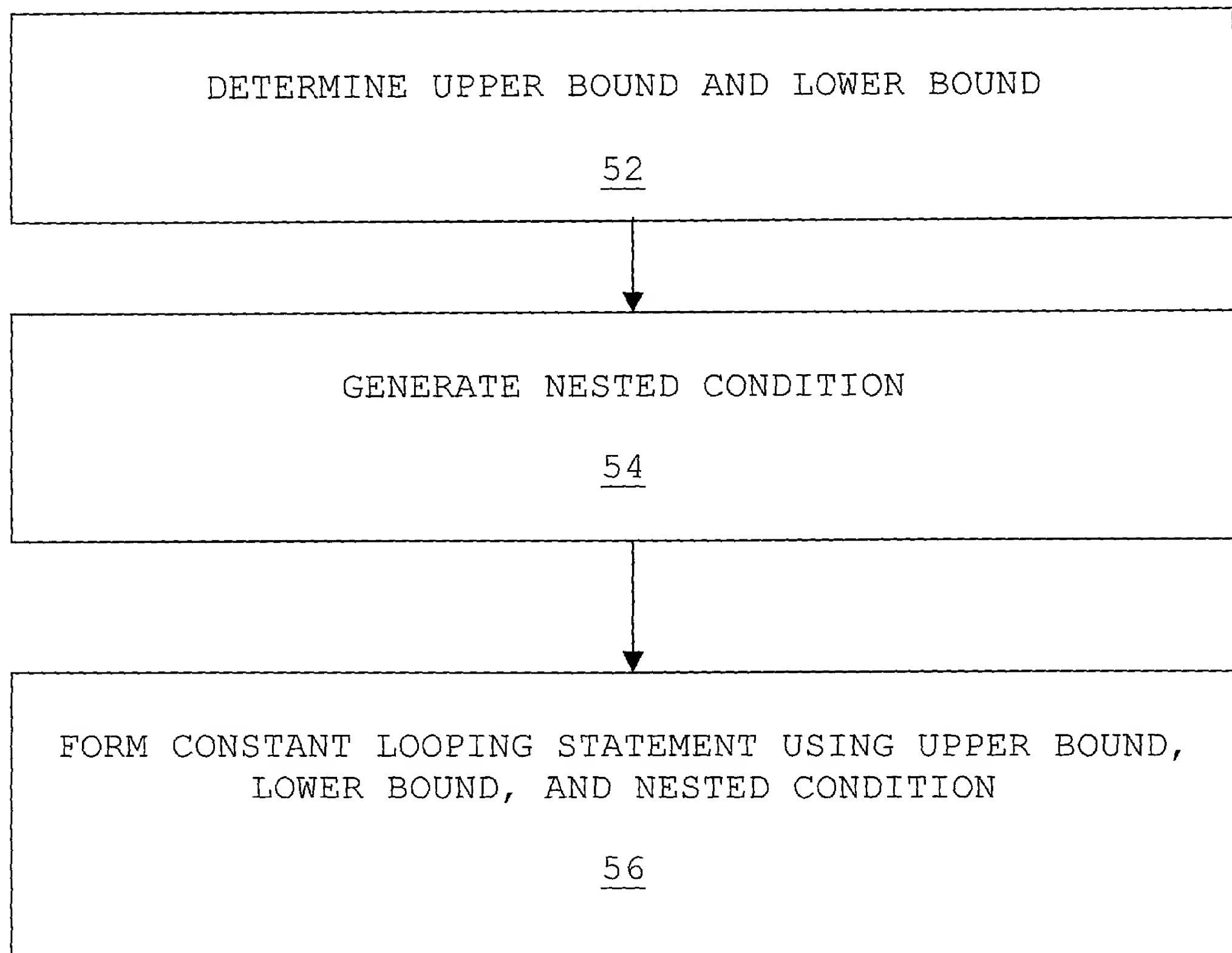


FIGURE 5

60 { FOR (LOWER_BOUND_EXPRESSION; UPPER_BOUND_EXPRESSION;
INCREMENT_EXPRESSION) 66
if (INIT && EXIT) 68
STATEMENT_BODY 69

FIGURE 6

80
82
84
70
72
74
76
78

```
reg i [3:0];  
reg j [1:0];  
reg k [2:0];  
for (j<=i; i<k; i=i+1)  
    statement_body
```

FIGURE 7

92 94 96
for (m=i; m<7; m++)
If (j<=m && m<k) 98
statement_body 100
90

```
graph TD
    92 --> m_i["m=i"]
    94 --> m_lt_7["m<7"]
    96 --> m_inc["m++"]
    98 --> if_stmt["If (j<=m && m<k)"]
    100 --> stmt_body["statement_body"]
    90 --- code_block["for (m=i; m<7; m++) If (j<=m && m<k) statement_body"]
```

FIGURE 8

112
WHILE (x <= 15)

if (x <= y) 114

begin

fpl_bit[x+y] = mm_iru[x-y]; 116

end

110

FIGURE 9